

Laboratorio Electrónica Digital

Santiago Ramirez Arenas

Docente: José Alfredo Jaramillo Villegas

Universidad Tecnológica de Pereira

19 de marzo de 2021

Lógica Combinatoria

OBJETIVO

En este laboratorio se estudiarán dispositivos de lógica combinatoria implementando circuitos lógicos. Se diseñarán estos dispositivos usando diferentes enfoques haciendo un uso efectivo de las opciones sintácticas del lenguaje de descripción de hardware SystemVerilog.

PROCEDIMIENTO DEL LABORATORIO

Problemas de lógica combinatoria. En este laboratorio se deberá diseñar y verificar los siguientes circuitos lógicos usando el lenguaje de descripción de hardware SystemVerilog:

- Problemas encontrados durante el diseño con sus respectivas las soluciones.
- Captura de pantalla del diseño.
- Captura de pantalla del banco de pruebas.
- Captura de pantalla de la simulación.
- Interpretación de los resultados de la simulación.
- Conclusiones generales

SOLUCIÓN

Circuito combinatorio de 4 a 1 bits.

o Simplificar por mapas de Karnaugh.

o Implementar usando una ecuación booleana.

o Implementar un llamado a un módulo de compuerta lógica

Tabla de verdad ejercicio uno

A	B	C	D	S
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Mapa de Karnaugh ejercicio uno

AB/CD	0.0	0.1	1.1	1.0
0.0	1	0	1	0
0.1	0	1	1	0
1.1	1	0	1	0
1.0	1	0	0	0

Diseño ejercicio uno mediante ambos métodos

```

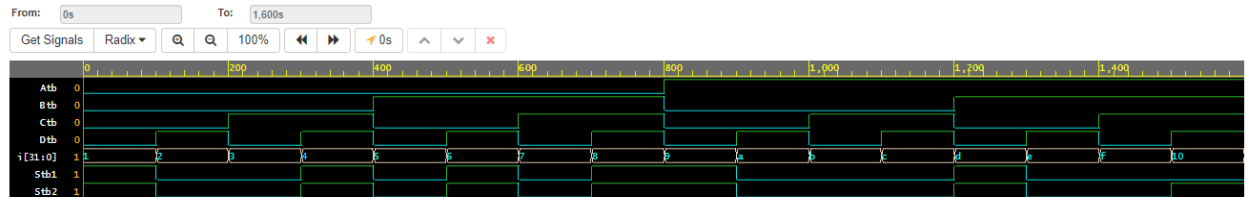
1 module circuito_UNO(
2   input logic A,B,C,D,
3   output logic S1,S2);
4   wire U,V,W,X,Y;
5
6   assign S1= (~B & ~C & ~D) | ( A & ~C & ~D) | (~A & B & D) | (~A & C &
   D);
7
8   and (U,!B, !C,!D);
9   and (V,A,!C,!D);
10  and (W,!A,B,D);
11  and (X, !A,C,D);
12  and (Y, B,C,D);
13  or (S2,U,V,W,X,Y);
14
15 endmodule

```

Testbench ejercicio uno

```
1 module test_circuito_UNO ();
2
3   logic Atb=0;
4   logic Btb=0;
5   logic Ctb=0;
6   logic Dtb=0;
7   logic Stb1;
8   logic Stb2;
9   integer i=1;
10
11   circuito_UNO circuito (
12     .A(Atb),
13     .B(Btb),
14     .C(Ctb),
15     .D(Dtb),
16     .S1(Stb1),
17     .S2(Stb2));
18
19   initial begin
20     $dumpfile("dump.vcd");
21     $dumpvars(1,test_circuito_UNO);
22     for(i=1;i<17;i=i+1)
23       begin
24         #100
25         if(i==8) begin
26           Atb=1;
27         end
28         if(i%4==0) begin
29           Btb=~Btb;
30         end
31         if(i%2==0) begin
32           Ctb=~Ctb;
33         end
34         Dtb=~Dtb;
35       end
36     $finish;
37   end
38
39 endmodule
```

Simulación ejercicio uno



Circuito lógico 5 a 1 bits.

- o Hacer tabla de verdad del circuito.
- o Simplificar por mapas de Karnaugh.
- o Implementar usando una ecuación booleana.
- o Implementar un llamado a un módulo de compuerta lógica.

Según la siguiente función lógica $f(a, b, c, d, e) = (0, 1, 2, 3, 6, 7, 8, 9, 10, 13, 15, 17, 19, 21, 23, 28, 29, 30)$

Tabla de verdad ejercicio dos

A	B	C	D	E	S
0	0	0	0	0	1
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	0
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	0

Mapa de Karnaugh ejercicio dos

AB/CDE	0.0.0	0.0.1	0.1.1	0.1.0	1.1.0	1.1.1	1.0.1	1.0.0
0.0	1	1	1	1	1	1	0	0
0.1	1	1	0	1	0	1	1	0
1.1	0	0	0	0	1	0	1	1
1.0	0	1	1	0	0	1	1	0

Diseño por ambos métodos ejercicio dos

```

1 module circuito_DOS(
2     input  logic A,B,C,D,E,
3     output logic S1,S2);
4     wire S,T,U,V,W,X,Y;
5
6     assign S1 = (~A & ~B & D) | (~A & ~C & ~E) | (A & B & C & ~E) | (~A &
       ~C & ~D) | (~A & B & C & E) | (A & ~B & E) | (B & C & ~D & E);
7
8     and (S,!A,!C,!D);
9     and (T,!A,!C,!E);
10    and (U,!A,!B,D);
11    and (V,A,!B,E);
12    and (W,!A,C,D,E);
13    and (X,B,C,!D,E);
14    and (Y,A,B,C,!E);
15    or (S2,S,T,U,V,W,X,Y);
16
17 endmodule

```

Testbench ejercicio dos

```

1 module test_circuito_DOS;
2
3     logic Atb=0;
4     logic Btb=0;
5     logic Ctb=0;
6     logic Dtb=0;
7     logic Etb=0;
8     logic Stb1;
9     logic Stb2;
10    integer i=1;
11
12    circuito_DOS circuito (
13        .A(Atb),
14        .B(Btb),
15        .C(Ctb),
16        .D(Dtb),
17        .E(Etb),
18        .S1(Stb1),
19        .S2(Stb2));
20
21    initial begin
22        $dumpfile("dump.vcd");
23        $dumpvars(1,test_circuito_DOS);
24        for(i=1;i<33;i=i+1)
25            begin
26                #100
27                if(i==16) begin
28                    Atb=1;
29                end
30                if(i%8==0) begin
31                    Btb=~Btb;
32                end
33                if(i%4==0) begin
34                    Ctb=~Ctb;

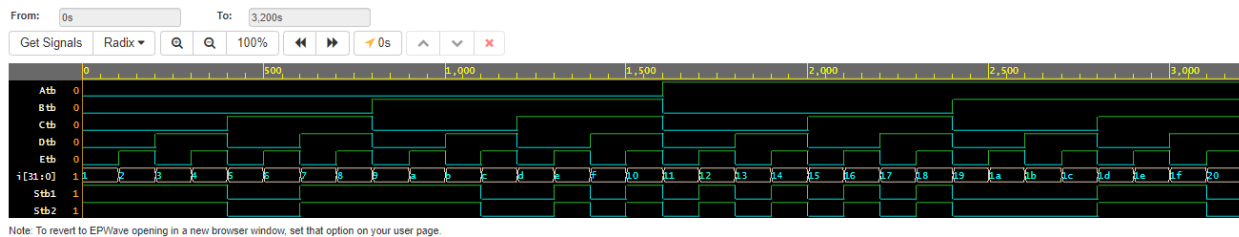
```

```

35     end
36     if (i%2==0) begin
37         Dtb=~Dtb;
38     end
39     Etb=~Etb;
40 end
41 $finish;
42 end
43
44 endmodule

```

Simulación ejercicio dos



- Circuito de una bombilla de 2 entradas.
- o Hacer tabla de verdad del circuito.
 - o Implementar usando una ecuación booleana.
 - o Implementar un llamado a un módulo de compuerta lógica

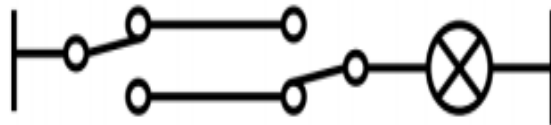


Tabla de verdad ejercicio tres

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

Diseño por ambos métodos ejercicio tres

```

1 module circuito_TRES(
2     input logic A,B,
3     output logic S1,S2);
4     wire X,Y;
5

```

```

6  assign S1 = (~A & ~B) | (A & B);
7
8  and (X,!A,!B);
9  and (Y, A,B);
10 or (S2,X,Y);
11
12 endmodule

```

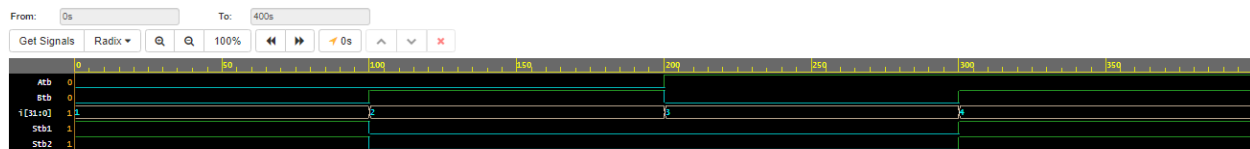
Testbench ejercicio tres

```

1 module test_circuito_TRES;
2
3  logic Atb=0;
4  logic Btb=0;
5  logic Stb1;
6  logic Stb2;
7  integer i=1;
8
9  circuito_TRES circuito (
10     .A(Atb),
11     .B(Btb),
12     .S1(Stb1),
13     .S2(Stb2));
14
15  initial begin
16     $dumpfile("dump.vcd");
17     $dumpvars(1,test_circuito_TRES );
18     for(i=1;i<5;i=i+1)
19         begin
20             #100
21             if(i==2) begin
22                 Atb=1;
23             end
24             Btb=~Btb;
25         end
26     $finish;
27 end
28
29 endmodule

```

Simulación ejercicio tres



Circuito combinatorio de 4 a 2 bits

Mediante dos bombas (m1 y m2) se controla el nivel de un depósito. El depósito tiene dos boyas (b1 y b2). Cuando el nivel está por debajo de la boya el contacto correspondiente está abierto. Las bombas sacan agua de dos pozos. Si no hay agua en el pozo la bomba no funciona. Para controlar esto, cada pozo lleva un sensor (n1, n2).

- Si el nivel del depósito supera la boya b1, las bombas están paradas.
- Si el nivel del depósito está entre la boya b1 y la b2, funciona la bomba m1, si hay agua suficiente en el pozo 1. Si no hay agua en el pozo 1 pero la hay en el 2, funciona la bomba m2.
- Si el nivel del depósito está por debajo de la boya b2, se activa la bomba m2, además de la m1.

o Hacer tabla de verdad del circuito.

o Implementar usando una ecuación booleana.

o Implementar usando un modelo comportamental.

Tabla de verdad ejercicio cuatro

B1	B2	N1	N2	M1	M2
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Mapa de karnaugh ejercicio cuatro

Mapa para M1

B1B2/N1,N2	0.0	0.1	1.1	1.0
0.0	0	0	1	1
0.1	0	0	1	1
1.1	0	0	0	0
1.0	0	0	0	0

$M1 = B1' \cdot N1$

Mapa para M2

B1B2/N1,N2	0.0	0.1	1.1	1.0
0.0	0	1	1	0
0.1	0	1	0	0
1.1	0	0	0	0
1.0	0	0	0	0

$M2 = (B1' \cdot B2' \cdot N2) + (B1' \cdot N1' \cdot N2)$

Diseño Ecuaciones Booleanas ejercicio cuatro

```

1 module circuito_CUATRO(
2   input logic B1,B2,N1,N2,
3   output logic M1,M2);
4

```



```

5  assign M1 = (~B1 & N1);
6  assign M2 = (~B1 & N2 & (~N1 | ~B2));
7
8  endmodule

```

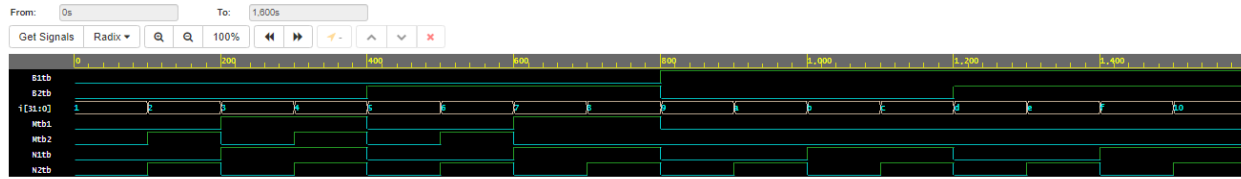
Testbench ejercicio cuatro

```

1  module  test_circuito_CUATRO;
2
3      logic B1tb=0;
4      logic B2tb=0;
5      logic N1tb=0;
6      logic N2tb=0;
7      logic Mtb1;
8      logic Mtb2;
9      integer i=1;
10
11  circuito_CUATRO circuito (
12      .B1(B1tb),
13      .B2(B2tb),
14      .N1(N1tb),
15      .N2(N2tb),
16      .M1(Mtb1),
17      .M2(Mtb2));
18
19  initial begin
20      $dumpfile("dump.vcd");
21      $dumpvars(1,test_circuito_CUATRO);
22      for(i=1;i<17;i=i+1)
23          begin
24              #100
25              if(i==8) begin
26                  B1tb=1;
27              end
28              if(i%4==0) begin
29                  B2tb=~B2tb;
30              end
31              if(i%2==0) begin
32                  N1tb=~N1tb;
33              end
34              N2tb=~N2tb;
35          end
36      $finish;
37  end
38
39  endmodule

```

Simulación ejercicio cuatro Ecuaciones Booleanas



Diseño Modelo Comportamental ejercicio cuatro

```

1 module circuito_CUATROMC(
2     input logic B1,
3     input logic B2,
4     input logic N1,
5     input logic N2,
6     output logic M1,
7     output logic M2
8 );
9
10
11 always @ (B1,B2,N1,N2)
12     begin
13         if (N1 == 0 && N2 == 0 )
14             begin
15                 M1 <= 0;
16                 M2 <= 0;
17             end
18         else begin
19             if (B1)
20                 begin
21                     M1 <= 0;
22                     M2 <= 0;
23                 end
24             else begin
25                 if (B2 && N1)
26                     begin
27                         M1 <= 1;
28                         M2 <= 0;
29                     end
30                 else begin
31                     if ((B2 && N1 && N2) || (N1 && N2))
32                         begin
33                             M1 <= 1;
34                             M2 <= 1;
35                         end
36                     else begin
37                         if (N2)
38                             begin
39                                 M1 <= 0;
40                                 M2 <= 1;
41                             end
42                     else begin
43                         if (N1)
44                             begin
45                                 M1 <= 1;

```

```

46         M2 <= 0;
47     end
48 end
49 end
50 end
51 end
52 end
53 end
54 endmodule

```

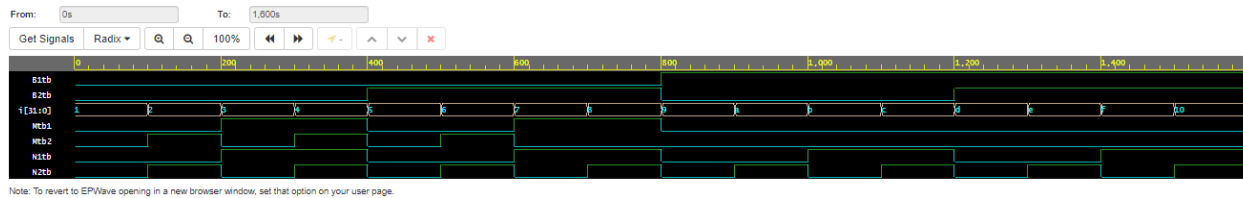
Testbench Modelo Comportamental ejercicio cuatro

```

1 module test_circuito_CUATROMC;
2
3     logic B1tb=0;
4     logic B2tb=0;
5     logic N1tb=0;
6     logic N2tb=0;
7     logic Mtb1;
8     logic Mtb2;
9     integer i=1;
10
11     circuito_CUATROMC circuito (
12         .B1(B1tb),
13         .B2(B2tb),
14         .N1(N1tb),
15         .N2(N2tb),
16         .M1(Mtb1),
17         .M2(Mtb2));
18
19     initial begin
20         $dumpfile("dump.vcd");
21         $dumpvars(1,test_circuito_CUATROMC);
22         for(i=1;i<17;i=i+1)
23             begin
24                 #100
25                 if(i==8) begin
26                     B1tb=1;
27                 end
28                 if(i%4==0) begin
29                     B2tb=~B2tb;
30                 end
31                 if(i%2==0) begin
32                     N1tb=~N1tb;
33                 end
34                 N2tb=~N2tb;
35             end
36         $finish;
37     end
38
39 endmodule

```

Simulación Modelo Comportamental ejercicio cuatro



Procedimiento del laboratorio

Procedimiento Ejercicio uno:

Inicialmente reconocemos el número de entradas y salidas. posteriormente organizamos el mapa de Karnaugh de 4 variables, con los respectivos 1 y 0 según corresponda, nos quedaría la siguiente ecuación lógica con la cual se realiza el diseño del circuito mediante Ecuaciones Booleanas y Llamado a módulo de compuertas lógicas

```
1 S = (~B & ~C & ~D) | (A & ~C & ~D) | (~A & B & D) | (~A & C & D);
```

Procedimiento Ejercicio dos :

Para el ejercicio número dos se tienen 5 entradas y una salida, las cuales son descritas por la función lógica

$$f(a, b, c, d, e) = (0, 1, 2, 3, 6, 7, 8, 9, 10, 13, 15, 17, 19, 21, 23, 28, 29, 30)$$

Teniendo en cuenta las combinaciones que nos brinda la función lógica asignamos los 1 de salida, los valores faltantes se complementan con los 0

Una vez que tenemos nuestra tabla de verdad procedemos con la estructura del Mapa de Karnaugh de 5 variables, con los respectivos 1 y 0 según corresponda, nos quedaría la siguiente ecuación lógica con la cual se realiza el diseño del circuito mediante Ecuaciones Booleanas y Llamado a módulo de compuertas lógicas

```
1 S = (~A & ~B & D) | (~A & ~C & ~E) | (A & B & C & ~E) | (~A & ~C & ~D) |
  (~A & B & C & E) | (A & ~B & E) | (B & C & ~D & E);
```

2

Procedimiento Ejercicio tres:

Para el ejercicio número tres, el cual es un circuito de una bombilla de dos entradas, vamos a establecer en 1 cuando el interruptor está arriba y en 0 cuando el interruptor está abajo, para la salida S, si la bombilla está encendida se le asigna un 1 y por el contrario si se encuentra en 0 está apagada. Teniendo presente los enunciados anteriores creamos nuestra tabla de verdad, de la cual podemos deducir la siguiente ecuación Booleana

```
1 S = (~A & ~B) | (A & B);
```

2

Nota: Esta operación lógica también puede ser representada como una operación XNOR (A XNOR B)

Procedimiento Ejercicio cuatro:

En el último circuito nos encontramos con 4 entradas 2 salidas, representando un sistema de riego, con dos pozos los cuales tienen dos sensores N1 y N2, Dos bombas M1 y M2, un depósito con las boyas B1 y B2.

Si las boyas marcan que el nivel está por encima de la marca se indica con un 1, de lo contrario se indica con un 0

Si los sensores indican que hay agua en el pozo se asigna un 1, de lo contrario se asigna un 0.

Si las bombas están encendidas se indica con un 1, de lo contrario se indica con un 0

- Si el nivel del depósito supera la boya b1, las bombas están paradas. Se representa cuando B1 y B2 están en 1, por consiguiente las bombas M1 Y M2 están en 0

- Si el nivel del depósito está entre la boya b1 y la b2, funciona la bomba m1, si hay agua suficiente en el pozo 1. Si no hay agua en el pozo 1 pero la hay en el 2, funciona la bomba m2. Se representa cuando B1 está en 0 y B2 en 1

- Si el nivel del depósito está por debajo de la boya b2, se activa la bomba m2, además de la m1.

```
1 M1 = (~B1 & N1);  
2 M2 = (~B1 & N2 & (~N1 | ~B2));  
3
```

Para la elaboración del Modelo Comportamental utilicé como guía

<https://es.wikibooks.org/wiki/Programacion>

Problemas encontrados durante el diseño con sus respectivas las soluciones.

Trabajar con mapas de Karnaugh cuando e tienen 5 variables puede resultar algo complejo

La implementación del modelo comportamental me resultó un poco complicado debido a que no tenía antecedentes de dicho modelo

Conclusiones generales.

El modelo comportamental es bastante útil para describir ciertas situaciones, siendo un poco parecido a la programación funcional.

Este laboratorio es útil para el planteamiento de problemas los cuales necesiten una serie de combinaciones lógicas para establecer ciertas salidas.

de _____ a enci_____